

Part Three: Visualization

Backend Platform

The backend platform is the server part that handles the application's business logic, data storage, and interaction with the frontend. The backend is typically implemented using various programming languages and frameworks such as Python, Java, Ruby, and PHP. **Its primary responsibility is to ensure correct data storage, processing, and transmission.**

In simple terms, the backend supports and handles the systems and technologies that run applications behind the scenes. **It manages servers, databases, and application logic, ensuring the content and features presented to front-end users function correctly.** **A backend platform typically includes servers, applications, databases, and APIs.**

Main Functions of the Backend

1. Data Storage and Retrieval: This is a basic function of the backend. It interacts with the database to store and retrieve data. In the sensor data logging project, two tables are used: Measurements and Sensors. **The Sensors table stores basic information about sensors,** including sensor ID and location. **The Measurements table stores the measurement data from sensors,** such as time, temperature, and humidity. The backend uses SQL statements to operate on the data stored in the relational database.

2. User Authentication and Authorization: Ensures that only legitimate users can access and manipulate system resources.

- **User Authentication:** Verifies user identity, commonly using a username and password.
- **User Authorization:** Determines what resources and actions authenticated users can access and perform, involving various roles such as **admin, regular user, and guest,** each with different permissions like view, edit, and assign permissions.

3. Business Logic Processing: **This is the core of the application, defining how the application processes data and performs operations.** In the sensor data project, business logic includes data processing, data validation, etc.

Business rules involve data recording. When a sensor uploads new measurement data, the system needs to save and process this data. Data processing may include calculating temperature and humidity change rates, comfort index, and visualizing data in charts.

Data validation ensures data integrity and accuracy, preventing incorrect data from entering the system. It includes form validation, checking required fields, correct formats, and ensuring consistency, like associating measurement data with a valid sensor ID.

API (Application Programming Interface)

The API is the bridge for data exchange between the frontend and backend. Through the API, the front-end can send requests, and the back-end returns the corresponding data.

RESTful API is commonly used, leveraging HTTP methods such as GET, POST, PUT, and DELETE.

- **GET:** Used to retrieve resources, e.g., getting information of all sensors or measurement data of a specific sensor.
- **POST:** Used to create resources, e.g., adding new sensors and measurement data.
- **PUT:** Used to update resources, e.g., updating sensor information or measurement data.
- **DELETE:** Used to delete resources, e.g., deleting a specific sensor or measurement data.

Common MySQL Commands

- **mysql -u root -p:** **Command to log in to the MySQL database.** The system will prompt you to enter the user's password.
- **mysql --version:** Check the MySQL version.